

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-44216

(43)公開日 平成9年(1997)2月14日

(51) Int.Cl. <sup>6</sup> G 05 B 19/05 F 02 D 45/00	識別記号 376	府内整理番号 F I G 05 B 19/05 F 02 D 45/00	技術表示箇所 A 376 B
---	-------------	---	----------------------

審査請求 未請求 請求項の数10 O.L (全 13 頁)

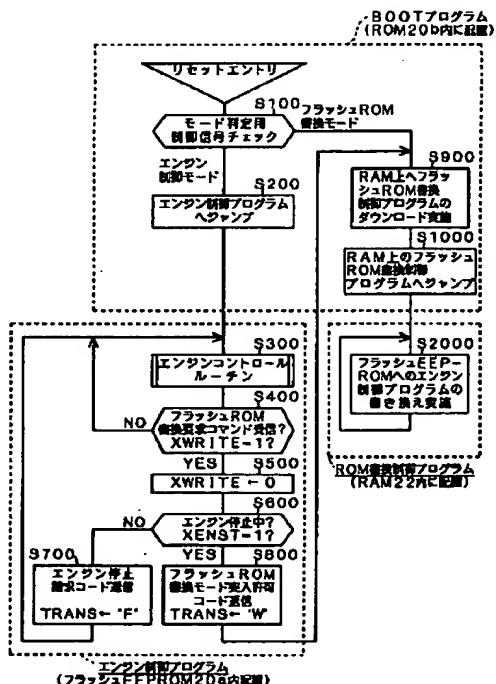
(21)出願番号 特願平7-195519	(71)出願人 株式会社デンソー 愛知県刈谷市昭和町1丁目1番地
(22)出願日 平成7年(1995)7月31日	(72)発明者 林 新之助 愛知県刈谷市昭和町1丁目1番地 日本電装株式会社内
	(72)発明者 柴田 浩 愛知県刈谷市昭和町1丁目1番地 日本電装株式会社内
	(74)代理人 弁理士 足立 勉

(54)【発明の名称】 機械制御装置

(57)【要約】

【課題】 プログラム等の書き換え時に機械が駆動状態であっても予期せぬ駆動状態を招かずにプログラム等の書き換えに入ることができる機械制御装置。

【解決手段】 エンジンが回転していると判定されると(S 600)、メモリ書換機へエンジン停止請求がなされ(S 700)、エンジンコントロールルーチン(S 300)は書き換えない。操作者が、エンジンを停止して、再度、メモリ書換機から書換要求をした場合には、ステップS 800の次にメモリ書換機から書換制御プログラムを受信し(S 900)、それを実行する(S 1000, S 2000)。この後、リセットにより新たなエンジンコントロールルーチン(S 300)が起動する。したがって、書き換え時にはエンジンは停止し起動時にも停止しているので、制御プログラム・データを変更しても自動車の機構が予期せぬ駆動状態とならない。



**【特許請求の範囲】**

【請求項1】内蔵プログラムおよび内蔵データにより機械の制御を行う機械制御装置であって、前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされていない条件下に、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行することを特徴とする機械制御装置。

【請求項2】前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされているときには、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行せず、前記機械の駆動がなされていることを報知することを特徴とする請求項1記載の機械制御装置。

【請求項3】前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされているときには、前記機械の駆動を停止する停止制御の後に、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行することを特徴とする請求項1記載の機械制御装置。

【請求項4】前記停止制御による前記機械の駆動の停止が、徐々に行われることを特徴とする請求項3記載の機械制御装置。

【請求項5】前記内蔵プログラムまたは内蔵データの一部または全部を、前記新たなプログラムまたはデータに書き換える機能を果たす書換プログラムを、書き換え対象でないメモリ領域に格納していることを特徴とする請求項1～4のいずれか記載の機械制御装置。

【請求項6】前記内蔵プログラムまたは内蔵データの一部または全部を前記新たなプログラムまたはデータに書き換える機能を果たす書換プログラムを外部から書き換え対象でないメモリ領域に読み込んで起動させる読込起動プログラムを、書き換え対象でないメモリ領域に格納していることを特徴とする請求項1～4のいずれか記載の機械制御装置。

【請求項7】前記機械が自動車であり、前記内蔵プログラムが自動車を制御する制御プログラムまたはその制御プログラムに関連するプログラムであり、前記内蔵データが自動車の制御に用いられる制御データまたはその制御データに関連するデータであることを特徴とする請求項1～6のいずれか記載の機械制御装置。

【請求項8】前記内蔵プログラムまたは内蔵データが、前記自動車のエンジンを制御するためのプログラムまたはデータであることを特徴とする請求項7記載の機械制御装置。

【請求項9】前記機械の駆動の有無が、前記自動車のエンジンの回転の有無にて判断されることを特徴とする請

求項7または8記載の機械制御装置。

【請求項10】前記内蔵プログラムおよび内蔵データが、書き換え可能なROMに格納されていることを特徴とする請求項1～9のいずれか記載の機械制御装置。

**【発明の詳細な説明】**

**【0001】**

【発明の属する技術分野】本発明は、機械の制御を行う機械制御装置の内蔵プログラムまたは内蔵データの書き換えに関し、例えば、自動車のエンジン制御のためのプログラムやデータの書き換えに関する。

**【0002】**

【従来の技術】従来、市場において、自動車のエンジンを制御しているエンジン制御装置におけるプログラムの書き換えは、プログラムの書換装置をそのエンジン制御装置に接続して、エンジン制御装置側の処理と書換装置側の処理との連携により実行していた。

【0003】すなわち、エンジン制御装置のCPUをリセットした直後に、そのCPU側の処理にて、プログラムの書換装置が输出するモード判定用制御信号ラインの信号状態をチェックし、プログラム書換モードでなければ、内蔵されている制御プログラムの起動に移り、エンジン制御を開始する。もし、モード判定用制御信号ラインの信号状態が、プログラム書換モードとなっていれば、内蔵の制御プログラムは起動せず、プログラムの書換装置から送信されて来る新しい制御プログラムを読み込んで、その内容を内蔵している制御プログラムの位置に書き込んだ後、その新しい制御プログラムを起動させていた。

【0004】リセット直後にこのようなプログラムの書き換え処理を行うのは、例えば、書き換え対象が制御プログラムであり、その制御プログラムが、仮に、書き換え中の異常動作（例えば電源ダウン）により、正しく実行されることのできない状態で終了した場合でも、再び書換モードに入りてその制御プログラムを正しく書きなおすことを可能とするためである。

【0005】このような制御プログラムの書き換えを、制御プログラムの実行前にしなくてはならないという制限を解決するものとして、特開平6-272611号公報に記載されたエンジン制御装置が存在する。このエンジン制御装置は、プログラムの書換装置からシリアル通信機能(SCI)割込みがかかった場合に、所定のシーケンスに従い、ROM書換プログラムをROM上からRAM上にコピーし、ROM上のSCI割込みのベクターアドレスをRAM上のROM書換プログラムへのアドレスに書き換えた後、ベクターアドレス以外のROM領域を、プログラムの書換装置から受信した新たな制御プログラムに書き換え、その後、ベクターアドレスも新しい内容に書き換えていた。

【0006】このことにより、制御プログラムの実行中であっても、制御プログラムを書き換えることができ

た。しかも、この書き換えはエンジン自体が駆動中でも可能としていた。

#### 【〇〇〇7】

【発明が解決しようとする課題】前記従来例では、エンジン駆動中において制御プログラムの書き換えをした場合にも、エンジンの現状に応じて設定されている制御用データは一旦RAMに退避しているので、そのデータを、プログラム等の書き換え後に元に戻せば、その後もエンジンは安定駆動するかのように考えられる。

【〇〇〇8】しかし、制御プログラムが変更されたということは、今までの制御とは異なる制御に突然に移行することを意味しており、エンジンが駆動しているとそのエンジンとともに、そのエンジンにより駆動されている自動車の各機構の駆動状態が急激に変化する恐れがあり、このような変化を予期していない作業者に不安感を生じさせる恐れがあった。

【〇〇〇9】このことは、制御プログラムがエンジン制御であることに関らず、例えば、ブレーキ制御、トランスマッション制御、サスペンション制御等においても、その制御プログラムや制御データの書き換え時に、駆動状態が急激に変化するという同じ問題が考えられた。

【〇〇10】本発明は、内蔵プログラムや内蔵データの書き換え時に機械が駆動装置により駆動されている場合にも、機械あるいはその駆動装置の急激な変化や制御不能等の予期せぬ駆動状態を招くこと無く、機械内蔵の制御プログラムや制御データの書き換えに入ることができる機械制御装置を提供することを目的とする。

#### 【〇〇11】

【課題を解決するための手段及び発明の効果】請求項1記載の発明は、内蔵プログラムおよび内蔵データにより機械の制御を行う機械制御装置であって、前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされていない条件下に、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行することを特徴とする機械制御装置である。

【〇〇12】請求項2記載の発明は、前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされているときには、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行せず、前記機械の駆動がなされていることを報知することを特徴とする請求項1記載の機械制御装置である。

【〇〇13】請求項3記載の発明は、前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされているときには、前記機械の駆動を停止する停止制御の後に、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送

信される新たなプログラムまたはデータに書き換える処理に移行することを特徴とする請求項1記載の機械制御装置である。

【〇〇14】請求項4記載の発明は、前記停止制御による前記機械の駆動の停止が、徐々に行われることを特徴とする請求項3記載の機械制御装置である。請求項5記載の発明は、前記内蔵プログラムまたは内蔵データの一部または全部を、前記新たなプログラムまたはデータに書き換える機能を果たす書換プログラムを、書き換え対象でないメモリ領域に格納していることを特徴とする請求項1～4のいずれか記載の機械制御装置である。

【〇〇15】請求項6記載の発明は、前記内蔵プログラムまたは内蔵データの一部または全部を前記新たなプログラムまたはデータに書き換える機能を果たす書換プログラムを外部から書き換え対象でないメモリ領域に読み込んで起動させる読込起動プログラムを、書き換え対象でないメモリ領域に格納していることを特徴とする請求項1～4のいずれか記載の機械制御装置である。

【〇〇16】請求項7記載の発明は、前記機械が自動車であり、前記内蔵プログラムが自動車を制御する制御プログラムまたはその制御プログラムに関連するプログラムであり、前記内蔵データが自動車の制御に用いられる制御データまたはその制御データに関連するデータであることを特徴とする請求項1～6のいずれか記載の機械制御装置である。

【〇〇17】請求項8記載の発明は、前記内蔵プログラムまたは内蔵データが、前記自動車のエンジンを制御するためのプログラムまたはデータであることを特徴とする請求項7記載の機械制御装置である。

【〇〇18】請求項9記載の発明は、前記機械の駆動の有無が、前記自動車のエンジンの回転の有無にて判断されることを特徴とする請求項7または8記載の機械制御装置である。請求項10記載の発明は、前記内蔵プログラムおよび内蔵データが、書き換え可能なROMに格納されていることを特徴とする請求項1～9のいずれか記載の機械制御装置である。

【〇〇19】ここで、請求項1の機械制御装置は、内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされていない条件下に、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行する。このことにより、機械が駆動されていると、内蔵プログラムまたは内蔵データの書き換えがなされないので、操作者に予期せぬ駆動状態となることが防止できる。

【〇〇20】前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされているときには、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行せ

ず、前記機械の駆動がなされていることを報知することとすれば、操作者に機械が駆動されているために、内蔵プログラムや内蔵データの書き換えがなされていないことを伝達することができるので、操作者は機械の駆動を停止した後、再度、内蔵プログラムや内蔵データを書き換えることにより、プログラムの書き換えが可能となる。

【0021】また、前記内蔵プログラムの起動中に外部から送信される書き換え命令があった場合に、前記機械の駆動がなされているときには、前記機械の駆動を停止する停止制御の後に、前記内蔵プログラムまたは内蔵データの一部または全部を、外部から送信される新たなプログラムまたはデータに書き換える処理に移行することとすれば、前述した操作者が行った機械の駆動停止処理も、自動的に行われると共に、引き続いてプログラムの書き換えも自動的に行われるので、機械が操作者に予期せぬ駆動状態となることを防止しつつ、プログラムの書き込み処理もできて、効率的である。

【0022】この停止制御は、前記機械の駆動を直ちに停止するのではなく、徐々にしても良い。特に、自動車のエンジンなどは急激に停止するのではなく、徐々に停止することがエンジンの耐久性あるいはエンジンにより駆動される自動車の各機構の耐久性の点から、また操作者を不安にさせないことからも好ましい。例えば、アイドル状態までエンジンの回転数を低下させた後に停止すれば良い。

【0023】尚、前記新たなプログラムまたはデータを、前記内蔵プログラムまたは内蔵データの一部または全部と書き換える機能を果たす書換プログラムを、書き換え対象でないメモリ領域に格納していることにより、書換プログラム自身の実行時に自分自身を書き換えることがなく、それ以降の処理に異常を生じることを防止できる。尚、前記書き換え対象でないメモリ領域は、例えば通常の書き換え不能なROMが挙げられる。また書き換え可能なROMやRAMであつても、その内容の書き換えが禁止されているならば、書き換え対象でないメモリ領域として用いることができる。

【0024】また、前記新たなプログラムまたはデータを前記内蔵プログラムまたは内蔵データの一部または全部と書き換える機能を果たす書換プログラムを外部から書き換え対象でないメモリ領域に読み込んで起動させる読み起動プログラムを、書き換え対象でないメモリ領域に格納していても良い。このように、書換プログラムも、最初から有しているのではなく、より簡単な読み起動プログラムを備えておき、前記内蔵プログラムまたは内蔵データを書き換える際に外部から読み込んで、それを起動して書き換えを行っても良い。プログラムを小さくすることができ、メモリの節約となる。尚、書換プログラムを読み込む書き換え対象でないメモリ領域は、RAMや書き換え可能なROMが挙げられる。読み起動プ

ログラムが格納されている書き換え対象でないメモリ領域は、通常の書き換え不能なROM、書き換え可能なROMあるいはRAMが挙げられる。

【0025】より具体的には、例えば、前記機械が自動車であり、前記内蔵プログラムが自動車を制御する制御プログラムまたはその制御プログラムに関連するプログラムであり、前記内蔵データが自動車の制御に用いられる制御データまたはその制御データに関連するデータである。したがって、自動車においても異常な駆動状態を招くこと無く、プログラムやデータの書き換えができる。制御プログラムに関連するプログラムとは、直接、制御には関連しないがタイマカウント処理といった制御に影響するプログラムなどが該当する。また制御データに関連するデータとは、直接、制御のデータとしては使用されないが、数値計算上のテーブルといった制御に影響するデータなどが該当する。

【0026】また、前記内蔵プログラムまたは内蔵データが、前記自動車のエンジンを制御するためのプログラムまたはデータであるとすると、エンジンの駆動状態にあるいはそのエンジンにより駆動されている自動車の各機構に、予期せぬ駆動状態を招くこと無く、プログラムやデータの書き換えができる。

【0027】この場合、例えば、前記機械の駆動の有無は、前記自動車のエンジンの回転の有無にて判断しても良い。したがって、エンジンが回転していれば、プログラムやデータの書き換えを禁止したり、エンジンが回転していることを報知したり、あるいはエンジンの回転を自動的に停止させてからプログラムやデータの書き換えを実行させることができる。この他、車速や変速機のニュートラルギア状態をチェックして、前記自動車の各機構の駆動の有無を判断しても良い。例えば、車速がゼロの場合に駆動されていないと判断したり、変速機のギアがニュートラルである場合に駆動されていないと判断する。

【0028】尚、前記内蔵プログラムおよび内蔵データは、例えば、書き換え可能なROMに格納される。この書き換え可能なROMとしては、フラッシュEEPROMあるいはEEPROMが一般的であるが、他の書き換え可能なROMでも良い。

#### 【0029】

#### 【発明の実施の形態】

【実施の形態1】図1は、本発明の一実施形態のエンジン制御装置2の全体構成を示すブロック図である。図1において、エンジン制御装置2は自動車に搭載され、内燃機関型エンジンの制御を行う。エンジンには、運転状態を表す様々なセンサ4が取り付けられており、これらセンサ4からの信号がエンジン制御装置2に入力される。エンジン制御装置2は、これらセンサ4からの信号を波形処理する入力処理回路6と、このセンサ信号を入力してエンジンの最適運転状態を演算するCPU8と、

このCPU8で演算された結果を制御信号として受けてエンジンに取付けられた燃料噴射装置や点火装置などのアクチュエータ10を駆動する出力回路12と、外部機器であるメモリ書換機14とのデータ通信用の通信回路16とを備えている。

【0030】CPU8には制御用プログラムに従い動作するマイクロ・プロセッサ・ユニット（以下、MPUという）18と、このMPU18を動作させるに必要なプログラムおよびデータを格納するROM20と、MPU18の演算結果を格納するRAM22と、前記入力処理回路6および通信回路16からの信号を受けるとともに、出力回路12に制御信号を出力するI/O24とが備えられている。なお、ROM20は、一旦書き込んだデータを全て消去することができ、しかも再書き込み可能なフラッシュEEPROM（通称、フラッシュROMまたはフラッシュメモリ）20aが一部に用いられている。他の一部は書き換え不可能なROM20bが用いられている。尚、この書き換え不可能なROM20bの代りに、フラッシュEEPROMやEEPROMが用いられても、消去や書き込みを許可しなければ良い。

【0031】尚、メモリ書換機14とエンジン制御装置2とは、通信バス接続用コネクタ26にて信号的に接続可能となっている。この通信バス接続用コネクタ26にて接続された通信用バスライン28はCPU8とメモリ書換機14との間でのシリアル通信機能によるデータ伝送を可能としている。また、この通信バス接続用コネクタ26により接続されたモード判定用制御信号ライン30は、後述するごとくリセット直後におけるCPU8の実行モードをメモリ書換機14側から設定するものである。

【0032】また、所定の条件下で、通信用バスライン28を介してCPU8が受信したプログラムやデータは、その一部がRAM22に格納されて起動対象となり、他の一部が書き換え可能なフラッシュEEPROM20a内に格納されていたプログラムあるいはデータの一部または全部と置き換る。

【0033】前記ROM20およびRAM22のメモリ領域の内容を図2に示す。フラッシュEEPROM20aには、エンジン制御プログラム・データ格納エリア32が設けられて、既に該当する制御プログラムおよび制御データが格納されている。すなわち内蔵プログラムおよび内蔵データとして存在している。

【0034】書き込み不能なROM20bには、ブートプログラム格納エリア34が設けられて、既に該当するブートプログラムが格納されている。また、RAM22にはフラッシュROM書換制御プログラム格納エリア36および演算用ワーク領域38が設けられ、所定条件下にメモリ書換機14から受信するフラッシュROM書換制御プログラムをフラッシュROM書換制御プログラム格納エリア36に格納すると共に、その実行時に演算作

業のために演算用ワーク領域38が使用される。

【0035】CPU8には、通常は、エンジン制御プログラム・データ格納エリア32にはエンジン制御プログラムおよびエンジン制御データが、更にブートプログラム格納エリア34にはブートプログラムが存在し、CPU8は、リセット直後にリセットエントリアドレスとして設定されているブートプログラムを起動した後、そのブートプログラムにてエンジン制御プログラムをコールして、エンジン制御を実行している。

【0036】ブートプログラムおよびエンジン制御プログラムのフローチャートを図3に示す。CPU8がリセットされると、まず、書き換え不能とされているROM20b内のブートプログラムが起動して、モード判定用制御信号ライン30の内容から、モード判定用制御信号をチェックする（S100）。ここでモード判定用制御信号のチェックにより、エンジン制御モードであると判定されると、エンジン制御プログラムへジャンプする（S200）。このことにより、フラッシュEEPROM20a内のエンジン制御プログラムの処理に移り、エンジン制御データを参照して行われるエンジンコントロールルーチンを実行する（S300）。

【0037】このエンジンコントロールルーチンは、例えば、図4のフローチャートに示すごとくである。すなわち、まず、エンジンストール判定処理（S110）がなされ、続いて、エンジン回転数算出処理（S120）、各種センサ入力処理（S130）、目標燃料噴射量演算処理（S140）、目標燃料噴射時期演算処理（S150）、目標燃料噴射圧演算処理（S160）、アクチュエータ制御パルス出力処理（S170）および各種自己診断処理（S180）が実行される。

【0038】前記エンジンストール判定処理（S110）は、エンジンに設けられた回転数センサからエンジンの回転数に応じて出力されるパルス信号を、所定時間X（sec）以上、CPU8が検出しなかったことによりエンジンストールであると判定する処理であり、図5のフローチャートに示すごとくである。

【0039】このエンジンストール判定処理において、まず、直前のパルス信号のタイミングから計時した、パルス信号を検出しない時間を表すタイムカウンタCENSTの値を判定して（S112）、CENSTが所定時間X以上の値であれば、エンジンストールであるとしてエンジンストールフラグXENSTをセットし（S114）、CENSTが所定時間X未満の値であれば、エンジンストールフラグXENSTをリセットする（S116）。

【0040】このような、タイムカウンタCENSTは、図6のフローチャートにより示す処理によりなされる。すなわち、図6（a）の時間周期割込処理によりタイムカウンタCENSTは、所定周期でインクリメント（S117）される。そして、図6（b）のエンジン回

転数パルス入力割込処理により、エンジン回転数パルスが入力する毎にタイムカウンタC E N S Tはクリア(S 118)される。したがって、エンジン回転数パルスがCPU8に入力しない限り、タイムカウンタC E N S Tは次第に増加することになり、その値はエンジン回転数パルスを検出しない時間を表すことになる。尚、図6(b)のエンジン回転数パルス入力割込処理では、同時に、エンジン回転数パルスの間隔に基づいてエンジン回転数演算のためのパルス幅計測処理(S 119)が行われている。

【0041】このエンジンコントロールルーチン(S 300)を一旦終了して、再度、エンジンコントロールルーチン(S 300)を繰り返す前に、フラッシュROM書換要求コマンドを受信したか否かが、フラッシュROM書換要求フラグXWR I T Eの状態にて判定される(S 400)。

【0042】このフラッシュROM書換要求フラグXWR I T Eは、エンジン制御装置2に接続されているメモリ書換機14から通信用バスライン28を介してシリアル信号を受信した場合に割込実行されるSCI割込処理内で設定される。すなわち、CPU8がメモリ書換機14からシリアル信号を受信すると、図7のフローチャートに示すSCI割込みが実行されて、まず、メモリ書換機14から書き換え命令としてのフラッシュROM書換要求コマンド'W'を受信したか否かが判定される(S 2010)。フラッシュROM書換要求コマンド'W'であれば、フラッシュROM書換要求フラグXWR I T Eをセットする(S 2020)。フラッシュROM書換要求コマンドでなければ、このまま処理を終了する。

【0043】したがって、エンジンコントロールルーチン(S 300)が繰り返すごとに、このフラッシュROM書換要求フラグXWR I T Eのセット状態がチェックされる(S 400)。フラッシュROM書換要求フラグXWR I T Eがセットされていなければ、直ちに、エンジンコントロールルーチン(S 300)を繰り返す。

【0044】フラッシュROM書換要求フラグXWR I T Eがセットされていた場合、まず、このフラッシュROM書換要求フラグXWR I T Eがリセットされる(S 500)。次にエンジンが停止しているか否かを、図5のエンジンストール判定処理にて処理されているエンジンストールフラグX E N S Tをチェックして判定する(S 600)。もしエンジンストールフラグX E N S Tがリセットされていれば、エンジンは回転していると判定されて、通信用バスライン28を介してメモリ書換機14への返信コードTRANSにエンジン停止請求コード'F'がセットされて(S 700)、エンジンコントロールルーチン(S 300)を繰り返す。

【0045】この返信コードTRANSにセットされたエンジン停止請求コード'F'は、CPU8の図示しない送信割込処理により、通信用バスライン28を介して

直ちにメモリ書換機14に送信される。このエンジン停止請求コード'F'を受信したメモリ書換機14は、そのエンジン停止請求コード'F'あるいはそのエンジン停止請求コード'F'の内容を示すコメントを、メモリ書換機14自身のディスプレイに表示する。

【0046】このことにより、操作者に対して、エンジン制御装置2側からエンジンの停止が要求される。したがって、エンジンが停止されない限り、メモリ書換機14からフラッシュROM書換要求コマンド'W'を出力していても、CPU8は、フラッシュROM書換処理(S 900以降)には移行せず、エンジンは、そのまま、今までの制御プログラム・制御データによる制御が継続される。

【0047】次に、操作者が、エンジンを停止して、再度、メモリ書換機14からフラッシュROM書換要求コマンド'W'を出力した場合には、エンジンの停止に伴いエンジンストールフラグX E N S Tがセットされるので、ステップS 600の次に、フラッシュROM書換モード突入許可コード'W'を返信コードTRANSにセットする処理がなされる(S 800)。次に、メモリ書換機14から、CPU8のRAM22上へのフラッシュROM書換制御プログラムのダウンロードを実行する(S 900)。

【0048】すなわち、フラッシュROM書換モード突入許可コード'W'を受信したメモリ書換機14は、所定のハンドシェイクの基で、メモリ書換機14にセットされているROM、RAMあるいは磁気ディスク等の記憶媒体から、フラッシュROM書換制御プログラムを通信用バスライン28を介して出力する。CPU8は、ステップS 900の処理にて、このフラッシュROM書換制御プログラムを受信してRAM22上のフラッシュROM書換制御プログラム格納エリア36に格納する。

【0049】フラッシュROM書換制御プログラムの受信が終了すれば、RAM22上のフラッシュROM書換制御プログラム格納エリア36の処理開始アドレスにジャンプして、フラッシュROM書換制御プログラムを起動する(S 1000)。フラッシュROM書換制御プログラム(S 2000)にては、次のような手順にてフラッシュEEPROM20aのエンジン制御プログラム・データ格納エリア32の全てあるいは一部のプログラム・データを書き換える。

【0050】1. メモリ書換機14から通信用バスライン28を介して、書き換える対象アドレス(あるいは消去対象ブロック)を読み込み、フラッシュEEPROM20a上の対象アドレスから、書き換えデータ分の領域(あるいは消去対象ブロック領域)を消去処理する。

【0051】2. メモリ書換機14から通信用バスライン28を介して、新たなプログラム・データを読み込んで、旧来のプログラムやデータを消去したフラッシュEEPROM20a上の領域に書き込む。

3. CPU8がメモリ書換機14から書き換えに必要な全てのプログラム・データを受信して、フラッシュEEPROM20aの書き換えが完了すれば、信用バスライン28を介してメモリ書換機14に書換完了信号を出力し、以後、この書換完了信号の出力を繰り返す無限ループに入る。

【0052】この書換完了信号を受信したことにより、メモリ書換機14のディスプレイには、書換完了のメッセージが表示される。したがって、操作者は、モード判定用制御信号ライン30をエンジン制御モードに設定して、リセット操作をCPU8側に行えば、ブートプログラム格納エリア34は、書き換えが禁止されているエリアなので、ブートプログラムは図3に示した旧来のプログラムが起動して、まず、モード判定用制御信号がチェックされる(S100)。メモリ書換機14からのモード判定用制御信号はエンジン制御モードに変更されているので、ステップS200にて、フラッシュEEPROM20aのエンジン制御プログラム・データ格納エリア32に格納されているエンジン制御プログラムを起動させる。

【0053】このエンジン制御プログラムあるいはエンジン制御プログラムにて利用される制御データは、前述したごとく全部またはその一部が書き換えられている。したがって、ステップS300のエンジンコントロールルーチンでは書き換え前とは異なるエンジン制御が開始されることになる。ただし、この時、エンジンは停止しているため、従来技術のごとく自動車の機構に予期せぬ駆動がなされることはない。

【0054】尚、ステップS400～ステップS800の部分はここでは書き換えないで、エンジン制御状態から、エンジン制御プログラム・データ格納エリア32の書き換え処理に移る機能は維持される。勿論、ステップS400～ステップS800を書き換えて同じ内容なら同じ機能が維持される。また同じ内容でなくとも、より改良された処理に書き換えて良い。

【0055】また、このステップS400～ステップS800の処理を、エンジン制御プログラム・データ格納エリア32のデータとするのではなく、ブートプログラムと同じく、書き換えが不可能（あるいはステップS2000の処理では書き換えが禁じられた）ROM20bに配置すれば、そのまま維持される。この場合も、新たなエンジン制御プログラム側では、周期的に処理を繰り返す間に、ステップS400～ステップS800の処理を実行するように設定されていれば良い。

【0056】本実施の形態では、エンジンが停止されていないと、エンジン制御プログラム・制御データの書き換え処理がなされないので、エンジン駆動中に制御プログラムや制御データが切り替わったことにより、自動車が予期せぬ駆動状態となることがない。このため、操作者に不安感を与えることがない。

【0057】また、書き換えた後のリセット後に、いきなりステップS300の処理から始まるのではなく、ブートプログラム格納エリア34に存在するブートプログラムから始まるので、ステップS2000にて書き換えに失敗して、エンジン制御プログラム・データ格納エリア32が消去状態のままとなった場合に、リセットからいきなり何も書かれていない処理に移るために生じるプログラムの異常な動作を防止できる。

【0058】尚、ステップS900およびステップS1000が読み起動プログラムに該当する。フラッシュROM書換制御プログラムが書換プログラムに該当する。

【実施の形態2】次に図3に示した処理と異なる処理の形態を、図8のフローチャートに示す。他の構成・処理については実施の形態1と同じなので説明は省略する。図8の処理が図3の処理と異なる点は、ステップS600にてエンジンストールフラグXENSTがリセットされていて、エンジンが回転中であると判定された場合に、ステップS700の代りにステップS720およびステップS740を実行する点である。

【0059】まず、ステップS720では、メモリ書換機14へエンジン強制停止を知らせるために、返信コードTRANSにエンジン強制停止コード'D' C'がセットされる。このことにより信用バスライン28を介してメモリ書換機14へエンジン強制停止が伝達される。したがって、このエンジン強制停止コード'D' C'を受信したメモリ書換機14は、エンジンが自動的に停止されることをディスプレイに表示して操作者に報知する。

【0060】ステップS740では、目標燃料噴射量をゼロにしたりディストリビュータへの電源供給停止により、エンジンを停止させる。このようにしエンジンが停止されるので、ステップS300に戻る必要が無くなり、ステップS740からは直ちにステップS900の処理に移る。以後、実施の形態1と同じく、エンジン制御プログラム・データ格納エリア32内のプログラム・データの書き換え処理に移る。

【0061】本実施の形態では、エンジン制御プログラム・制御データの書き換え処理をしようとすると、エンジンが自動的に停止されるので、エンジン駆動中にエンジン制御プログラム・制御データの書き換え処理がなされることは無い。したがって、エンジン駆動中に制御プログラムが別のプログラムに切り替わることはないので、自動車の駆動状態が予期せぬ状態となることが無く、操作者に不安感を与えることもない。他の効果は実施の形態1と同じである。

【0062】[その他] 前記各実施の形態において、ステップS2000は、書き換えが終了するとメモリ書換機14への書換完了信号の出力を繰り返す無限ループに入ったが、書き換えが終了したら、メモリ書換機14へ書換完了信号を出力した後、CPU8自身をリセットしても良い。また、リセットせずにステップS300の処

理にジャンプして、新しいプログラム・データによるエンジン制御を開始しても良い。この場合も、エンジンは駆動していないので問題ない。

【0063】前記実施の形態2において、ステップS720にてメモリ書換機14へエンジン強制停止を知らせた後、直ちにステップS740にてエンジンを停止していたが、メモリ書換機14へ知らせること無く、エンジン制御装置2側でランプ等を点滅させてエンジン停止することを報知した後、エンジンを停止しても良い。

【0064】また、前記実施の形態2において、エンジンを急激に停止すると、エンジンに支障が生じたり、操作者に不安感を与える恐れが有るので、一旦、アイドル状態にまで、エンジン回転数を低下させて、その後に停止させても良い。あるいは、ステップS720にてメモリ書換機14へエンジン強制停止を知らせた後、操作者がエンジンの強制停止を認識すると考えられる時間、エンジンの回転を維持した後、エンジンを停止しても良い。

【0065】また、自動的に停止させる場合も、ステップS720にてメモリ書換機14へエンジン強制停止の許可を求めて回答を待つようにし、その求めに対して、メモリ書換機14から操作者が停止許可のコードを送信して来た場合に、ステップS740を実行しても良い。操作者が停止不許可のコードを送信して来た場合には、エンジンを停止せずに、再度ステップS300の処理に戻り、プログラム・データの書き換え処理は中止する。

【0066】前記各形態において、リセット直後に、ステップS100にてモード判定用制御信号ライン30からのモード判定用制御信号をチェックして、フラッシュROM書換モードであれば書き換え処理に移行していくが、エンジン制御モードにおいてメモリ書換機14からの書き換え命令により同様の書き換え処理に移行できることから、特にステップS100の判定は必要ない。このことから、モード判定用制御信号ライン30は特に設けなくても、エンジン制御プログラム・制御データの書き換えは可能となる。

【0067】したがって、従来、例えば、エンジン制御装置2等からの自己診断データを、モード判定用制御信号ライン30のない通常の通信バス接続用コネクタを介して読み取るダイアグノーシス読取装置などの各種ツールに、メモリ書換機14の機能を組み込んで使用することができ、従来のツールを新たに必要としない利点が生じる。

【0068】前記各形態は、エンジン制御のプログラム・データの書き換えに限られていたが、本発明の適用範囲は、これに限られない。すなわち、例えば、ブレーキ制御、トランスマッション制御、サスペンション制御等においても、エンジンの駆動によりブレーキ機構、トランスマッション機構、サスペンション機構の全部または一部が駆動されている可能性が有り、そのような状況下

で、それらブレーキ制御、トランスマッション制御、サスペンション制御等の制御プログラムや制御データの書き換えを行うと、それぞれの駆動状態が予期せぬ変化をして、乗員に不安を生じさせる恐れが有るという問題、あるいは書き換えに失敗すると正常に作動しないというエンジン制御と同様な問題が存在するからである。

【0069】このため、エンジンが回転している場合には、実施の形態1, 2と同じく、ブレーキ制御プログラム、トランスマッション制御プログラムあるいはサスペンション制御プログラム等を書き換える命令があった場合に、操作者に警告して、エンジンを停止させたり、あるいは自動的にエンジンを停止させることにより、エンジンに駆動されている機構の予期せぬ駆動を生じることがない。

【0070】また、これらブレーキ制御プログラム、トランスマッション制御プログラムあるいはサスペンション制御プログラム等を書き換える場合も、更に前記各実施の形態の場合も、エンジン回転の有無でなく、他の条件として自動車の速度検出器のデータにより自動車が走行駆動しているか否かを判定して、自動車が走行駆動していないと判断される場合にプログラム・データの書き換えを許可することとしても良い。また、ギアがニュートラル状態にある場合に、自動車が走行駆動されていないとしてプログラム・データの書き換えを許可することとしても良い。

【0071】前記各実施の形態においては、メモリ書換機14からフラッシュROM書換制御プログラムを受信してRAM22上のフラッシュROM書換制御プログラム格納エリア36に格納してから、そのフラッシュROM書換制御プログラムを起動していたが、フラッシュROM書換制御プログラムを、メモリ書換機14から読み込むのではなく、書き換え不能な（または書き換え禁止）ROM20bに備えておけば、メモリ書換機14からダウンロードする必要はない。

【0072】前記各実施の形態においては、リセット時のスタートは、書き換え不能な（または書き換え禁止）ROM20bに備えられたブートプログラムにて行っている。これは、もしエンジン制御プログラムの書き換えに失敗した場合に、リセットから、いきなり何も書かれていらない領域にジャンプして、CPU8の動作が異常となることを防止するためであるが、更にステップS200のエンジン制御プログラムへのジャンプ処理において、そのジャンプ先に何も書かれていなことが判明すればジャンプを停止して、エラーコード等をメモリ書換機14に出力して操作者に知らせるようにしても良い。

#### 【図面の簡単な説明】

【図1】 本発明の一実施形態のエンジン制御装置の全体構成を示すブロック図である。

【図2】 メモリ領域の内容説明図である。

【図3】 実施の形態1のブートプログラムおよびエン

ジン制御プログラムのフローチャートである。

【図4】 エンジンコントロールルーチンのフローチャートである。

【図5】 エンジンストール判定処理のフローチャートである。

【図6】 タイムカウンタCENSTの設定処理であり、(a)は時間周期割込処理、(b)はエンジン回転数パルス入力割込処理である。

【図7】 フラッシュROM書換要求フラグXWRITをセットするSCI割込み処理のフローチャートである。

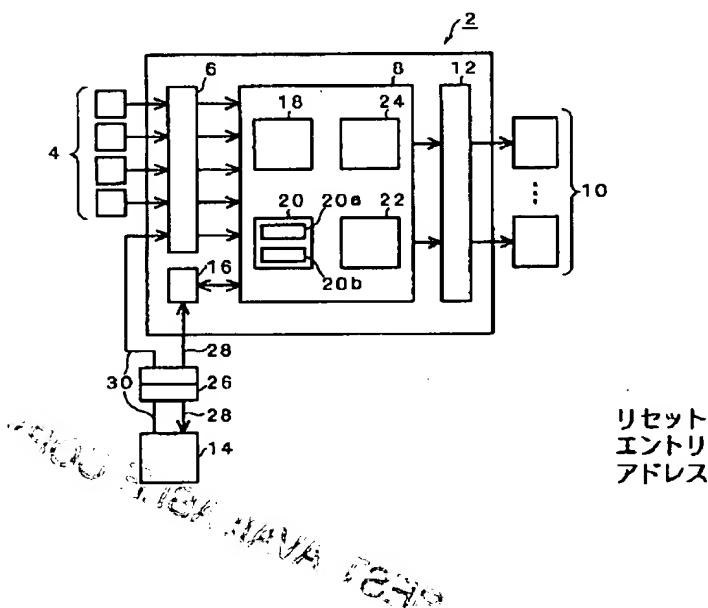
【図8】 実施の形態2のブートプログラムおよびエンジン制御プログラムのフローチャートである。

【符号の説明】

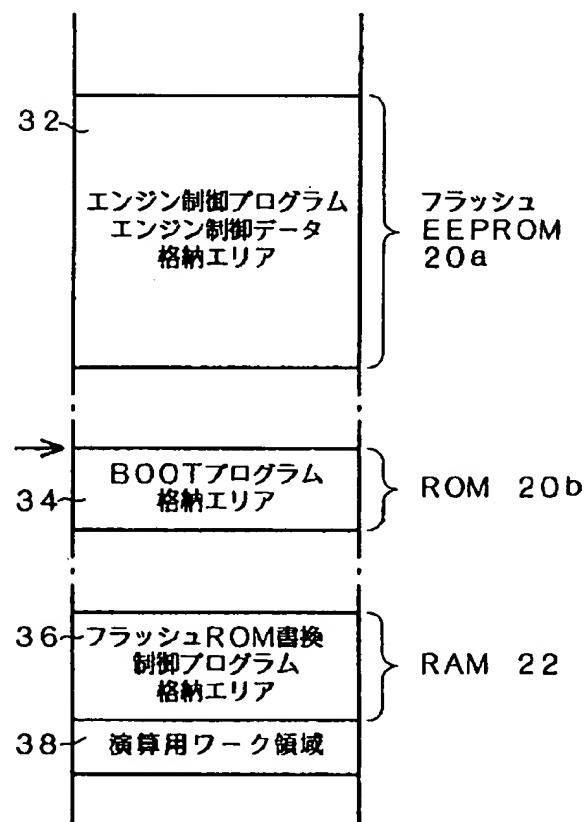
2…エンジン制御装置 4…センサ 6…入力  
処理回路

8…CPU 10…アクチュエータ 12…出力回路  
14…メモリ書換機 16…通信回路 18…MPU  
20…ROM 20a…フラッシュEEPROM  
20b…書き換え不可能な（または書き換えが禁止されている）ROM  
22…RAM 24…I/O 26…通信バス接続用コネクタ  
28…信用バスライン 30…モード判定用制御信号ライン  
32…データ格納エリア 34…ブートプログラム格納エリア  
36…フラッシュROM書換制御プログラム格納エリア  
38…演算用ワーク領域

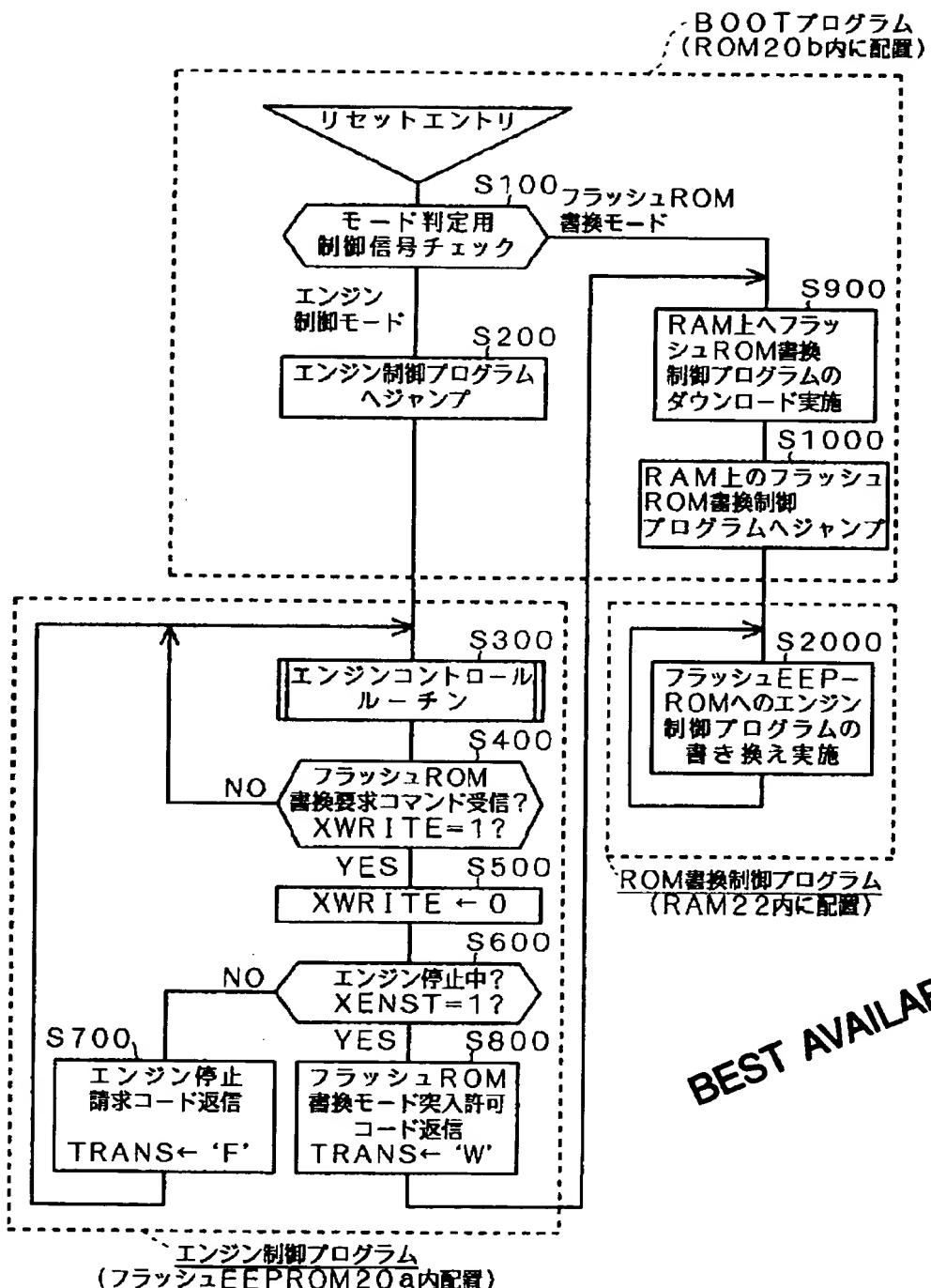
【図1】



【図2】

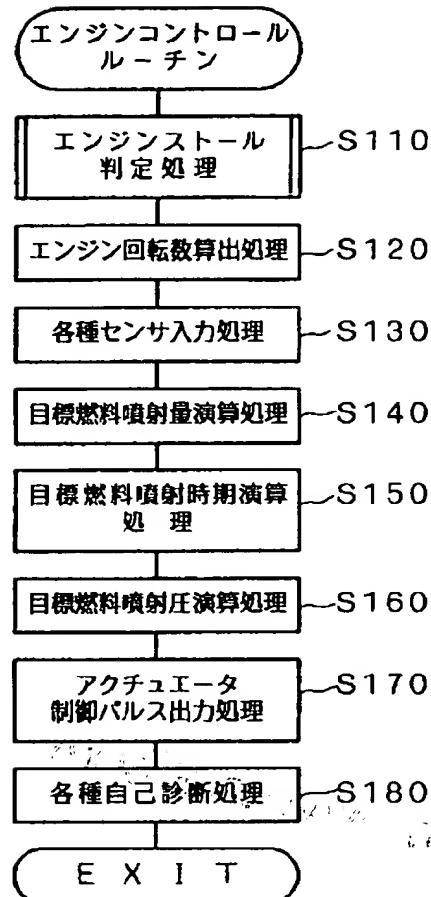


【図3】

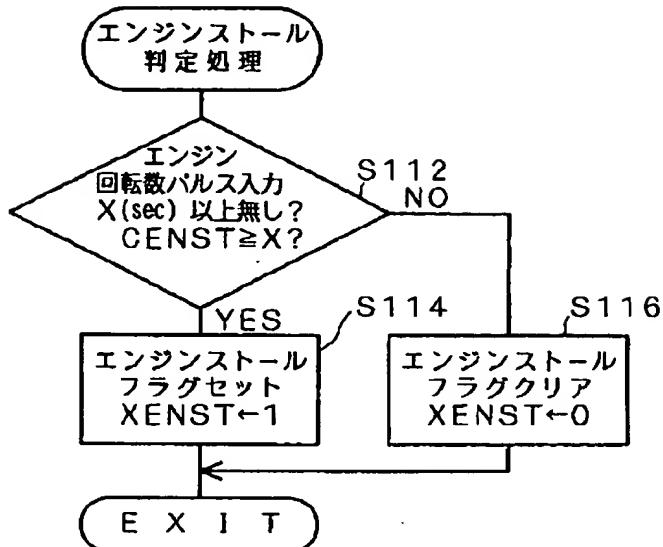


BEST AVAILABLE COPY

【図4】

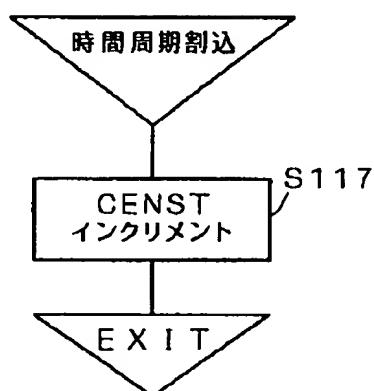


【図5】

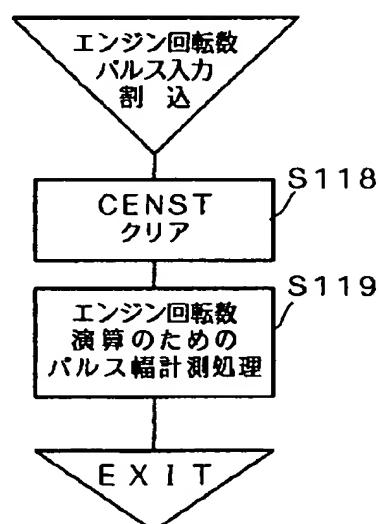


【図6】

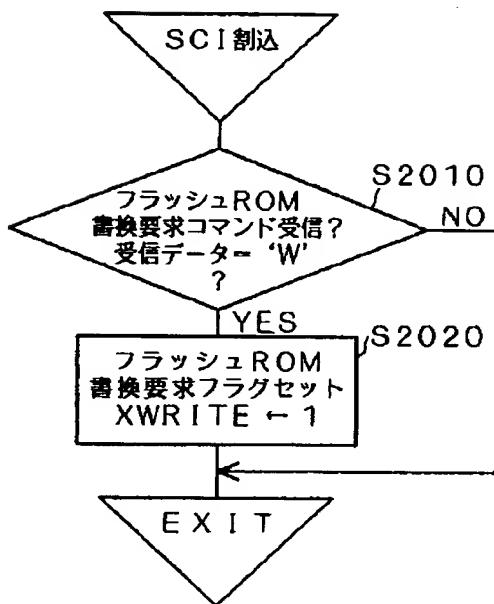
(a)



(b)

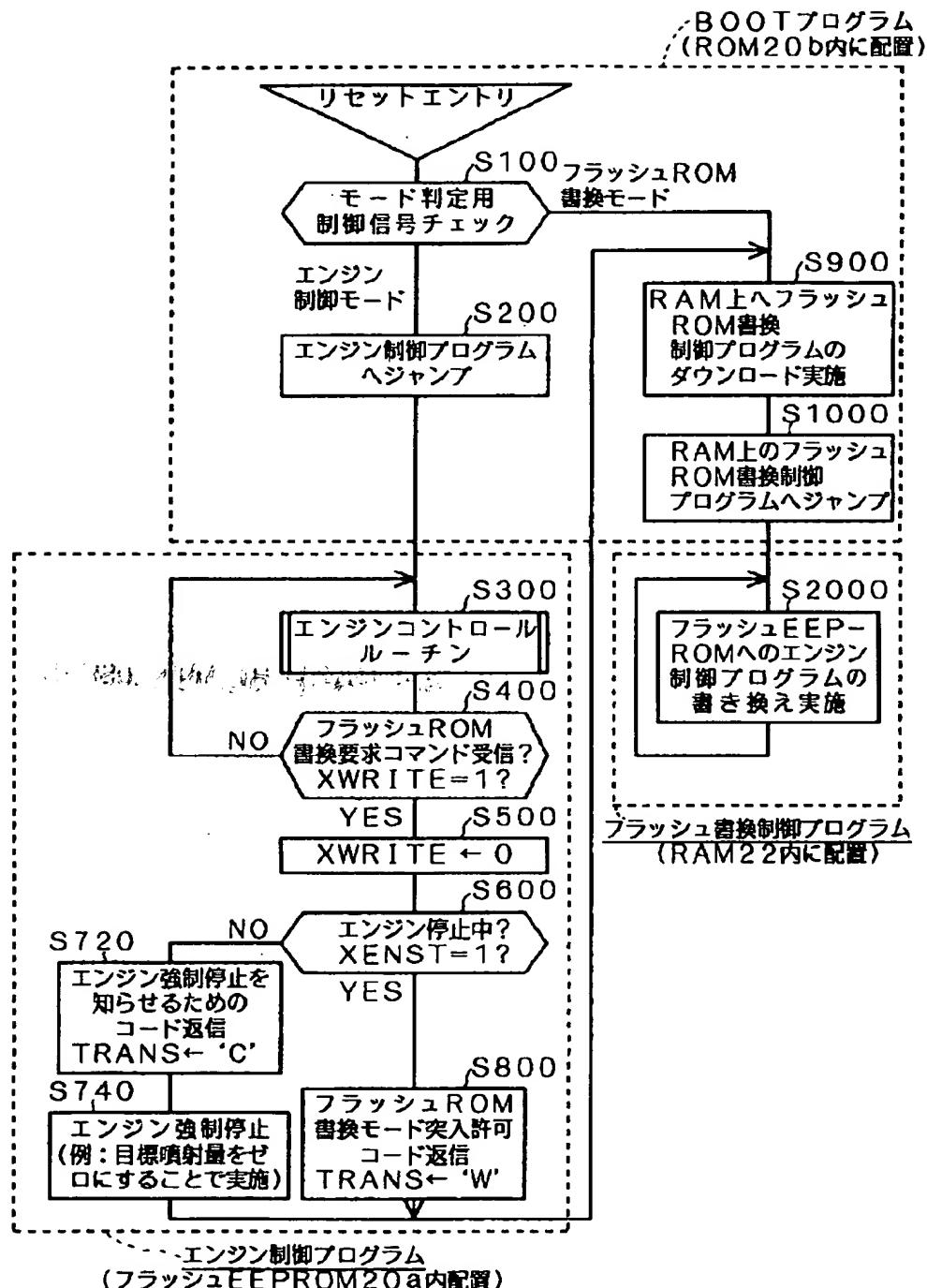


【図7】



BEST AVAILABLE COPY

【図8】



**THIS PAGE BLANK (USPTO)**